Amendments to the Claims

1. (Currently amended) A method comprising:

executing an exchange function comprising exchanging an address contained in a local variable ~~pointer~~ for an address contained in a global variable~~pointer~~;

jumping to ~~an~~ the address contained in the local variable ~~pointer~~ wherein~~,~~ the address in the local variable ~~pointer~~ points to the exchange function~~,~~ address or ~~the local pointer points~~ to ~~an address accessing~~ a shared resource address; and

executing the exchange function after accessing the shared resource.

2. (Currently amended) The method of claim 1 wherein the exchange function is a method of a class including a property comprising the global variable~~pointer~~.

3. (Original) A computer system comprising plural multi-tasking processes performing the method of claim 1.

4. (Original) A computer system comprising plural time-sharing threads performing the method of claim 1.

5. (Canceled)

6. (Currently amended) The method of claim 1 wherein the local variable ~~pointer~~ is associated with a process.

7. (Currently amended) The method of claim 1 wherein the local variable ~~pointer~~ is associated with a thread.

8. (Currently amended) The method of claim 1 wherein the local variable ~~pointer~~ is a property of a class associated with a thread.

9. (Currently amended) The method of claim 1 wherein the exchange function is a method of a class including a property comprising the local variable~~pointer, and the class is in a threads address space~~.

10. (Currently amended) The method of claim 1 wherein the exchange function is a method of a class including a property comprising the global variable~~pointer, and the class is in a processes address space~~.

11. (Original) Plural child threads of a process in contention for the shared resource, and performing the method of claim 1.

12. (Original) The method of claim 11 wherein child threads each include a local variable, and the process includes the global variable.

13. (Original) The method of claim 1 wherein the exchange function is executed as an atomic unit.

14. (Original) A computer-readable medium comprising instructions for performing the method of claim 1.

15. (Currently amended) A computer-readable medium including instructions for performing functions comprising:

creating a global variable initialized with an address of a shared resource~~ access address~~;

creating plural threads each including a local variable initialized with an address of an exchange function~~ address~~;

the exchange function, ~~for~~ being called by the plural threads to exchange an address in a ~~exchanging a threads~~ calling thread's local variable with an address in the global variable;

after a thread calls the exchange function, ~~control flow for~~ transferring control to an address contained in ~~a~~ the thread's local variable wherein the address in the local variable points to the address of the exchange function or to the address of the shared resource; and

~~control flow for calling~~ the exchange function, being called by a thread after the thread ~~accessing~~ accesses the shared resource.

16. (Original) The computer-readable medium of claim 15 wherein the function for creating the global variable further comprises creating as an atomic unit.

17. (Original) The computer-readable medium of claim 15 wherein the shared resource is a memory location.

18. (Original) The computer-readable medium of claim 15 wherein the exchange function further comprises calling and returning from a sleep function before exchanging variables.

19. (Currently amended) The computer-readable medium of claim 15 wherein ~~during execution of the method by a created thread, control flow for~~ transferring control to the address contained in the thread's local variable, transfers control to one of the exchange function or the shared resource and execution of the thread continues at the transferred location.

20. (Currently amended) A computer system comprising:

a processor coupled to memory comprising a global variable;

a shared resource;

the global variable initialized with an address of the shared resource;

an exchange function for exchanging contents of the global variable with contents of local variables;

plural units of execution each comprising, ~~sharing the processor and performing a method comprising,~~

an associated local variable initialized with an address of the exchange function,

code for calling the exchange function to exchange contents of the global variable with contents of its associated local variable,

code for, after calling the exchange function, ~~executing an exchange function comprising changing a local pointer for a global pointer;~~ jumping to an address contained in its associated local variable ~~the local pointer~~ wherein~~,~~ the ~~local pointer~~ address points to the address of the exchange function~~,~~ or to the ~~local pointer points to an~~ address of the ~~accessing a~~ shared resource, ~~;~~ and

code for executing the exchange function after accessing the shared resource.

21. (Currently amended) The computer system of claim 20 wherein said plural units of execution are threads, and a first thread includes code for, after executing the exchange function, jumping ~~jumps~~ to the address accessing the shared resource, and before the first thread exits the shared resource, the system includes code for transferring control of the processor ~~transfers~~ to a second thread, and the second thread includes code for, upon executing the exchange function, jumping ~~jumps~~ to the address of and cycles in the exchange function.

22. (Currently amended) The computer system of claim 21 wherein after control transfers to the first thread, and after the first thread exits the shared resource and executes the exchange function, and after control of the processor transfers to the second thread, the second thread includes code for ~~executes~~ executing the exchange function and ~~jumps~~ jumping to the address accessing the shared resource.

23. (Currently amended) The computer system of claim 20 wherein the exchange function is a method of a class including a property comprising the global variable ~~pointer~~.

24. (Canceled)

25. (Original) The computer system of claim 20 wherein the units of execution are processes.